LISTING OF THE CLAIMS

Claims 1-28 are pending. Please amend claims 1, 4, 9, 12, 17, 20, 25 and 28. No claims are added or canceled. This listing of claims replaces all prior versions and listings of claims in the application.

1. (Currently amended) A method comprising:

generating a set of Steiner trees and paths from an undirected graph of vertices representing terminal and Steiner nodes; ~~and~~

merging the Steiner trees and the paths to produce linked and edge-disjoint S-Steiner trees such that if a subset S of the vertices is k edge-connected, then there are $\alpha_{|S|}k$ edge-disjoint Steiner trees for S, where α_s is at minimum a sequence that tends to an asymptotic approximation factor of $|S|/4$ as s tends to infinity; and

utilizing the linked and edge-disjoint S-Steiner trees for a practical application of multiple applications, the multiple applications comprising data multicasting in a network to present information to a set of users and determining a VLSI circuit design to share an electric signal between terminals.

2. (Original) A method as recited in claim 1, wherein generating further comprises analyzing an undirected graph of vertices representing terminal and Steiner nodes to produce a Steiner Tree between two terminal nodes of the terminal nodes, the two terminal nodes now being processed nodes.

3. (Original) A method as recited in claim 1, wherein generating further comprises:

for each unprocessed vertex of the vertices, identifying one or more respective paths from the unprocessed vertex to each of a set of processed terminal vertices of the vertices to inductively grow the undirected graph by creating the Steiner trees, the unprocessed vertex now being a processed vertex;

for each unprocessed vertex of the vertices, identifying one or more respective paths from the unprocessed vertex to each of a set of processed terminal vertices of the vertices to inductively grow the undirected graph by creating the Steiner trees, the unprocessed vertex now being a processed vertex; and

for each Steiner tree:

determining if a path of the paths shares an edge with the Steiner tree; and

responsive to determining that the path shares the edge, shortcutting the path to a vertex of the Steiner tree by removing a portion of the path that is subsequent to the edge, each Steiner tree being used to shortcut a path of the paths being a path-tree.

4. **(Currently amended)** A method as recited in claim 1, wherein <u>the application is data multicasting in a network, and wherein</u> the vertices represent respective sending, receiving, and router network nodes, and wherein the method further comprises:

receiving a set of requests for streaming data from at least a subset of vertices of *S*, the at least a subset representing receiving network nodes;

identifying one or more of the edge-disjoint Steiner trees that comprise each of the at least a subset; and

multicasting the streaming data to the at least a subset over communication pathways identified by the one or more of the edge-disjoint Steiner trees.

5. **(Currently amended)** A method as recited in claim 1, wherein the substantially $\alpha_{|S|}k$ edge-disjoint Steiner trees for *S* are at minimum the following:

$$SOL_p(x) = x_0 + x_p - \left\lceil (x_0 - \sum_{i=p+1}^{s} ix_i)/p \right\rceil + \sum_{i=1}^{p-1} x_i$$
$$= \left\lfloor \frac{p-1}{p} x_0 + \sum_{i=1}^{p} x_i + \sum_{i=p+1}^{s} \frac{i}{p} x_i \right\rfloor,$$

wherein x_0 represents the Steiner trees not used to shortcut any path, $\cancel{x_0}$ <u>x_p</u> represents Steiner trees used to shortcut a ~~path,~~ <u>path.</u>

6. **(Original)** A method as recited in claim 5, wherein *p* is a number such that $\sum_{i=p+1}^{s} ix_i < x_0 \le \sum_{i=p}^{s} ix_i$.

7. **(Original)** A method as recited in claim 5, wherein if $x_0 \le sx_s$, $p = s$.

8. **(Original)** A method as recited in claim 5, wherein if $\sum_{i=1}^{s} ix_i < x_0$, $p = 0$.

9. **(Currently amended)** A computer-readable medium comprising computer-executable instructions for packing Steiner trees, the computer-executable instructions comprising instructions for:

generating a set of Steiner trees and paths from an undirected graph of vertices representing terminal and Steiner nodes; ~~and~~

merging the Steiner trees and the paths to produce linked and edge-disjoint S-Steiner trees such that if a subset S of the vertices is k edge-connected, then at minimum there are $\alpha_{|S|}k$ edge-disjoint Steiner trees for S, where α_s is a sequence that tends to an asymptotic approximation factor of $|S|/4$ as s tends to infinity; and

utilizing the linked and edge-disjoint S-Steiner trees for a practical application of multiple applications, the multiple applications comprising data multicasting in a network to present information to a set of users, and creating a VLSI circuit design to share electrical signal(s) between terminal nodes.

10. **(Original)** A computer-readable medium as recited in claim 9, wherein the computer-executable instructions for generating further comprise instructions for analyzing an undirected graph of vertices representing terminal and Steiner nodes to produce a Steiner Tree between two terminal nodes of the terminal nodes, the two terminal nodes now being processed nodes.

11. (Original) A computer-readable medium as recited in claim 9, wherein the computer-executable instructions for generating further comprise instructions for:

for each unprocessed vertex of the vertices, identifying one or more respective paths from the unprocessed vertex to each of a set of processed terminal vertices of the vertices to inductively grow the undirected graph by creating the Steiner trees, the unprocessed vertex now being a processed vertex;

for each unprocessed vertex of the vertices, identifying one or more respective paths from the unprocessed vertex to each of a set of processed terminal vertices of the vertices to inductively grow the undirected graph by creating the Steiner trees, the unprocessed vertex now being a processed vertex; and

for each Steiner tree:

determining if a path of the paths shares an edge with the Steiner tree; and

responsive to determining that the path shares the edge, shortcutting the path to a vertex of the Steiner tree by removing a portion of the path that is subsequent to the edge, each Steiner tree being used to shortcut a path of the paths being a path-tree.

12. (Currently amended) A computer-readable medium as recited in claim 9, wherein <u>the practical application is data multicasting in a network, and wherein</u> the vertices represent respective sending, receiving, and router network nodes, and wherein the computer-executable instructions further comprise instructions for:

receiving a set of requests for streaming data from at least a subset of vertices of *S*, the at least a subset representing receiving network nodes;

identifying one or more of the edge-disjoint Steiner trees that comprise each of the at least a subset; and

multicasting the streaming data to the at least a subset over communication pathways identified by the one or more of the edge-disjoint Steiner trees.

13. (Currently amended) A computer-readable medium as recited in claim 9, wherein the $\alpha_{|S|}k$ edge-disjoint Steiner trees for *S* are at minimum based on the following:

$$SOL_p(x) = x_0 + x_p - \left\lceil (x_0 - \sum_{i=p+1}^{s} ix_i)/p \right\rceil + \sum_{i=1}^{p-1} x_i$$
$$= \left\lfloor \frac{p-1}{p} x_0 + \sum_{i=1}^{p} x_i + \sum_{i=p+1}^{s} \frac{i}{p} x_i \right\rfloor,$$

wherein x_0 represents the Steiner trees not used to shortcut any path, ~~x_0~~ <u>x_p</u> represents Steiner trees used to shortcut a path.

14. (Original) A computer-readable medium as recited in claim 13, wherein p is a number such that $\sum_{i=p+1}^{s} ix_i < x_0 \leq \sum_{i=p}^{s} ix_i$.

15. **(Original)** A computer-readable medium as recited in claim 9, wherein if $x_0 \leq sx_s$, $p = s$.

16. **(Original)** A computer-readable medium as recited in claim 9, wherein if $\sum_{i=1}^{s} ix_i < x_0$, $p = 0$.

17. **(Currently amended)** A computing device comprising:

a processor;

a memory coupled to the processor, the memory comprising computer instructions executable by the processor for:

generating a set of Steiner trees and paths from an undirected graph of vertices representing terminal and Steiner nodes; ~~and~~

merging the Steiner trees and the paths to produce linked and edge-disjoint S-Steiner trees such that if a subset S of the vertices is k edge-connected, then at minimum there are $\alpha_{|S|}k$ edge-disjoint Steiner trees for S, where α_s is a sequence that tends to an asymptotic approximation factor of $|S|/4$ as s tends to infinity<u>; and</u>

<u>utilizing the linked and edge-disjoint S-Steiner trees for a practical application of multiple applications, the multiple applications comprising one or more of data multicasting in a network to present information to a set of users, and creating a VLSI circuit design to share electrical signal(s) between terminal nodes</u>.

18. (Original) A computing device as recited in claim 17, wherein the computer instructions for generating further comprise instructions for analyzing an undirected graph of vertices representing terminal and Steiner nodes to produce a Steiner Tree between two terminal nodes of the terminal nodes, the two terminal nodes now being processed nodes.

19. (Original) A computing device as recited in claim 17, wherein the computer instructions for generating further comprise instructions for:

for each unprocessed vertex of the vertices, identifying one or more respective paths from the unprocessed vertex to each of a set of processed terminal vertices of the vertices to inductively grow the undirected graph by creating the Steiner trees, the unprocessed vertex now being a processed vertex;

for each unprocessed vertex of the vertices, identifying one or more respective paths from the unprocessed vertex to each of a set of processed terminal vertices of the vertices to inductively grow the undirected graph by creating the Steiner trees, the unprocessed vertex now being a processed vertex; and

for each Steiner tree:

determining if a path of the paths shares an edge with the Steiner tree; and

responsive to determining that the path shares the edge, shortcutting the path to a vertex of the Steiner tree by removing a portion of the path that is subsequent to the edge, each Steiner tree being used to shortcut a path of the paths being a path-tree.

20. **(Currently amended)** A computing device as recited in claim 17, wherein the practical application is data multicasting in a network, and wherein the vertices represent respective sending, receiving, and router network nodes, and wherein the computer-executable instructions further comprise instructions for:

receiving a set of requests for streaming data from at least a subset of vertices of S, the at least a subset representing receiving network nodes;

identifying one or more of the edge-disjoint Steiner trees that comprise each of the at least a subset; and

multicasting the streaming data to the at least a subset over communication pathways identified by the one or more of the edge-disjoint Steiner trees.

21. **(Currently amended)** A computing device as recited in claim 17, wherein the $\alpha_{|S|}k$ edge-disjoint Steiner trees for S are at minimum based on the following:

$$SOL_p(x) = x_0 + x_p - \left\lceil (x_0 - \sum_{i=p+1}^{s} ix_i)/p \right\rceil + \sum_{i=1}^{p-1} x_i$$
$$= \left\lfloor \frac{p-1}{p} x_0 + \sum_{i=1}^{p} x_i + \sum_{i=p+1}^{s} \frac{i}{p} x_i \right\rfloor,$$

wherein x_0 represents the Steiner trees not used to shortcut any path, ~~x_0~~ $\underline{x_p}$ represents Steiner trees used to shortcut a ~~path,~~ path.

22. **(Original)** A computing device as recited in claim 21, wherein p is a number such that $\sum_{i=p+1}^{s} ix_i < x_0 \leq \sum_{i=p}^{s} ix_i$.

23. **(Original)** A computing device as recited in claim 21, wherein if $x_0 \leq sx_s$, $p = s$.

24. **(Original)** A computing device as recited in claim 21, wherein if $\sum_{i=1}^{s} ix_i < x_0$, $p = 0$.

25. **(Currently amended)** A computing device comprising:

means for generating a set of Steiner trees and paths from an undirected graph of vertices representing terminal and Steiner nodes; ~~and~~

means for merging the Steiner trees and the paths to produce linked and edge-disjoint S-Steiner trees such that if a subset S of the vertices is k edge-connected, then at minimum there are substantially $\alpha_{|S|}k$ edge-disjoint Steiner trees for S, where α_s is a sequence that tends to an asymptotic approximation factor of $|S|/4$ as s tends to infinity; and

means for implementing a practical application that utilizes the linked and edge-disjoint S-Steiner trees, the practical application comprising: (a) data multicasting in a network to present information to a set of users, or (b) creating a VLSI circuit design to share an electric signal between terminals.

26. **(Original)** A computing device as recited in claim 25, wherein the means for generating further comprise means for analyzing an undirected graph of vertices representing terminal and Steiner nodes to produce a Steiner Tree between two terminal nodes of the terminal nodes, the two terminal nodes now being processed nodes.

27. **(Original)** A computing device as recited in claim 25, wherein the means for generating further comprise:

means for each unprocessed vertex of the vertices, identifying one or more respective paths from the unprocessed vertex to each of a set of processed terminal vertices of the vertices to inductively grow the undirected graph by creating the Steiner trees, the unprocessed vertex now being a processed vertex;

for each unprocessed vertex of the vertices, means for identifying one or more respective paths from the unprocessed vertex to each of a set of processed terminal vertices of the vertices to inductively grow the undirected graph by creating the Steiner trees, the unprocessed vertex now being a processed vertex; and

for each Steiner tree:

means for determining if a path of the paths shares an edge with the Steiner tree; and

responsive to determining that the path shares the edge, means for shortcutting the path to a vertex of the Steiner tree by removing a portion of the path that is subsequent to the edge, each Steiner tree being used to shortcut a path of the paths being a path-tree.

28. (Currently amended) A computing device as recited in claim 25, wherein the application is data multicasting in a network, and wherein the vertices represent respective sending, receiving, and router network nodes, and further comprising:

means for receiving a set of requests for streaming data from at least a subset of vertices of *S*, the at least a subset representing receiving network nodes;

means for identifying one or more of the edge-disjoint Steiner trees that comprise each of the at least a subset; and

means for multicasting the streaming data to the at least a subset over communication pathways identified by the one or more of the edge-disjoint Steiner trees